

10/665,726

- 2 -

RECEIVED
CENTRAL FAX CENTER

FEB 28 2007

Amendments to the Specification

Please replace the paragraph at page 1, line 9 through line 18 with the following amended paragraph:

This application is also related to pending U.S. Patent Application No. 10/667,203, filed September 18, 2003, Application entitled "Field Oriented Pipeline Architecture for a Programmable Data Streaming Processor," (Attorney Docket No. 3336.1008-002); pending U.S. Patent Application No. 10/666,729, filed September 18, 2003, Application entitled "Asymmetric Streaming Record Data Processor Method and Apparatus," (Attorney Docket No. 3336.1016-001); pending U.S. Patent Application No. 10/668,113, filed September 18, 2003, entitled "Programmable Streaming Data Processor For Data Base Appliance Having Multiple Processing Unit Groups," (Attorney Docket No. 3336.1016-002); and pending U.S. Patent Application Serial No. 10/667,128, filed September 18, 2003 entitled "Programmable Data Streaming Architecture Having Autonomous and Asynchronous Job Processing Unit," (Attorney Docket No. 3336.1016-003), all of which are being filed together on the same date as this application.

Please replace the paragraph at page 5, line 5 through line 8 with the following amended paragraph:

More particularly, the PSDP is programmable by the host and/or CPU to interpret data in a specific format as it is read from the associated disk(s). This enables PSDP to perform portions of jobs directly, as it is read off the disk, prior to such data ever being forwarded to the CPU.

Please replace the paragraph at page 11, line 14 through 27 with the following amended paragraph:

The hosts 12 typically accept queries that are requests for data stored on mass storage devices, such as hard disk drives 23. Queries are typically provided in a format such as Structured Query Language (SQL), Open DataBase Connectivity (ODBC), Java DataBase Connectivity (JDBC), or the like. Hosts develop and optimize a query execution plan which typically consists of one or more jobs to be processed by other elements of the system. The requests may originate from any number of business applications that may reside on local processors [[28]]38, client computers 36, or separately running remote application software 30 that may access the host 12 through a computer network 33. The hosts 12 accept queries that can retrieve, modify, create and/or delete data stored on disk 23 and the schema for such data. The

10/665,726

- 3 -

hosts 12 also accept requests to start, commit, and rollback transactions against the data. The hosts 12 also perform typical administrative functions such as reporting on the status of the system 10, start and shutdown operation, backing up the current state of the data, restoring previous states of the data, replicating the data, and performing maintenance operations.

Please replace the paragraph at page 12, line 4 through 6 with the following amended paragraph:

Optionally, there is a load balancing function [[11]] 16 in front of the host 12 processors, which directs individual transactions to a specific host or hosts 12 so as to evenly distribute workload.

Please replace the paragraph at page 21, line 8 through line 26 with the following amended paragraph:

As shown more particularly in Fig. 4, a PSDP 28 consists of a finite state machine and related logic called the Data Engine 400 which implements filter logic and tuple generation; a CPU interface 404; a disk interface (here the ATA interface 408); a (read) memory First-In-First-Out (FIFO)/driver 406, a (read) disk FIFO/driver 407; and a write DMA FIFO/driver 402. The Data Engine 400 is located between the disk FIFO/driver 407 and the memory FIFO/driver 406. The PSDP 28 supports both a Programmed I/O (PIO) Mode-2 for register access by the CPU 26 and a UDMA (Ultra-Direct Memory Access) mode-4 for streaming data transfers. The terms "flow through" and "filtered" are used to differentiate UDMA read modes. Both the flow-through and filtered modes use the disk and memory FIFO/drivers. This allows the disk 23, memory [[28]] 27, and, if enabled, the filter logic 500 to process data at rates somewhat independent of each other. In flow-through mode, also referred to as "raw read" mode, data moves directly from the input to the output of the Data Engine 400 without being filtered. Data that is filtered has been processed, perhaps by culling records via a comparison and/or transaction ID processing (as described below), but always by reformatting the records into an output tuple, during which unselected fields can be dropped and PSDP-generated fields can be added. This process of culling records is called a "restrict" operation. The process of formatting fields into tuples is called a "project" (pronounced as in the operation of "throwing" something.)

10/665,726

- 4 -

Please replace the paragraph at page 26, line 6 through line 10 with the following amended paragraph:

More details of TID processing as performed by the TID processing block 510, includes rollback, are contained in ~~[[our]]~~ co-pending U.S. Patent Application No. 10/646,522 (~~Attorney Docket No. 3336.1017-001~~) ~~entitled "Controlling Visibility in Multi-Version Database Systems"~~ by ~~Foster D. Hinshaw et. al.~~, filed in the United States Patent and Trademark Office on August 22, 2003.

Please replace the paragraph at page 26, line 11 through line 19 with the following amended paragraph:

A detailed circuit diagram of the filter / comparison logic 500 is shown in Fig. ~~[[5]]~~ 6. The filter logic 500 supports up to 32 comparison (and hash) instructions. Each operates on a single field, which can be a record header field, the row address, or a data field. Multiple instructions can operate on fields whose nominal length does not exceed 16-bytes (fixed-length fields and short varchars), but only one instruction can operate on longer variable-length fields (varchars and BLOBs). The instruction results are combined and can be presented in the tuple as the Boolean results output, the up-to 32 instructions taken together with the transaction ID processing results determine the use/lose decision.

Please replace the paragraph at page 26, line 25 through line 26 with the following amended paragraph:

There is also a programmable, byte-wide substitution table 703 that can be used to simplify character field comparisons.